Seek new ideas even if out of the ordinary

- “To an uninspired scientist with a hammer, everything looks like a nail”
- If at first, the idea is not absurd, then there is no hope for it
  Albert Einstein

Seek new ideas even if you are unsure of their appeal

- “If we knew what it was we were doing, it would not be called research, would it?”

Wild Idea is a Sign of brilliance

- Imagination is more important than knowledge.
  - For knowledge is limited to all we now know and understand,
  - while imagination embraces the entire world, and all there ever will be to know and understand
- Logic will get you from A to B. Imagination will take you everywhere

... the Babbling of Old Fools?
History of Computation

5K YR Ago

Driven by necessity we have progressed from using fingers
to tables

to slide rules

M. A. El-Sharkawi, BIA

University of Washington

M. A. El-Sharkawi, BIA

to abacus

to mechanical machines

to more mechanical machines

to calculators
Impact of Modern Computing Power

- Staggering impact on our ability to
  - Compute through iterations
  - Create amazingly complex algorithms

New Wave of Computing

- Moore’s law is still true
  - Computers double their power every 18 month.
- Computationally intensive techniques hard to implement a few years ago are now feasible.
  - Among these techniques are the so-called biologically inspired algorithms (BIA).

Biologically Inspired Algorithms

Biologically Inspired Systems

- Philosophers argued that with all human achievements in science and engineering, nature still provides the best systems that can ever be fashioned.

- This is true even if we compare the most complex machine with the simplest form of a biological cell.
... And heeeere’s Albert

When the solution is simple, God is answering.

Albert Einstein

The ABC’s of BIA

BIA or Biocomputation

The use of biological processes or behavior as metaphor, inspiration, or enabler in developing new computing technologies

The field is highly multidisciplinary, Engineers, computer scientists, molecular biologists, geneticists, mathematicians, physicists, and others.

Main steps of Biologically Inspired Systems

Observe and study:
- Observe animal and human behaviors and Study biological structures
- Mimic:
  - Acquired knowledge may help us mimic nature and develop better engineering systems and machines.

BIA Systems

- Neural Networks
- Evolutionary Algorithms
- Fuzzy Systems
- Swarm Intelligence
- Boids
- Particle Swarm
- DNA Computing
- Artificial Life
- Intelligent Agents

Nature is a Powerful Paradigm

- Brain → Neural Networks
- Evolution theory → Evolutionary Algorithms
- Flocking birds → Particle Swarm Optimization, Boids
- Insects → Swarm Intelligence
- ......
**Why BIA?**

- May require little or no knowledge of the physical system they emulate.
- System is developed by observations and intuition.
- Can find substance in data bases through stochastic search.
- Can be self-tuned using raw experiential data.
- Noise tolerant and robust.
- Objectives no longer need to be in restrictive mathematical forms.
- Nonlinearity is no longer a disabling constraint.

**Key Improvements**

- The performance of BIAs is better understood and appreciated.
  - NN can have an explanation facilities and is no longer a "black boxes".
  - Stability criteria for fuzzy control, long lacking, can be established.
  - Convergence is improved and global optimization within a predetermined space can be achieved.

**Conventional Model based System**

- Sensors
- Model
- Parameter or State Estimation
- Model
- Decision
- Desired

**Non-Model based System**

- Data from Sensors
- Process Engine
- Decision
- Desired

**Classical Control: Design**

- System inputs
- Control Inputs
- Linear Model
- Objectives
- Constraints

**Classical Control: Operation**

- System inputs
- Control Inputs
- Actual System
- Objectives
- FEEDBACK
**Intelligent Control**

**Neural Networks**
**Input Patterns**

**Output Decision**

**Connections adjustment**

**Input layer**

**Hidden layer**

**Output layer**

---

**NN Training: El-Sharkawi’s Quintuplets**

---

**Testing of Trained NN**
Almost!

Evolutionary Algorithms

Evolutionary Algorithm vs SST

Single Search
Evolutionary Search

Population Pool

Individual

Evolutionary Algorithm

Fitness Evaluation
Crossover

Crossover point

#p 1 1 1 1 1 1 1 1
#q 0 0 0 0 0 0 1

Crossover

#p 1 1 1 1 1 1 1 1
#q 0 0 0 0 0 0 1

Success of Crossover

The only real valuable thing is intuition.

Albert Einstein

Experiment

Crisp Set/Subset

Universe (X)

Subset A

Subset B

Subset C

M. A. El-Sharkawi, BIA 55

M. A. El-Sharkawi, BIA 56

M. A. El-Sharkawi, BIA 57

M. A. El-Sharkawi, BIA 58

M. A. El-Sharkawi, BIA 59

M. A. El-Sharkawi, BIA 60
On a scale of one to 10, how good was the dive?

Close, heavy, light, big, small, smart, fast, slow, hot, cold, tall and short.

Example #1
- Billy has 9 toes. The probability Billy has 10 toes is zero.
- The fuzzy membership of Billy in the set of people with 10 toes is nonzero.

Example #2
- A bottle of liquid has a probability of $\frac{1}{2}$ of being rat poison and $\frac{1}{2}$ of being pure water.
- A second bottle’s contents, in the fuzzy set of liquids containing lots of rat poison, is $\frac{1}{2}$.
- The meaning of $\frac{1}{2}$ for the two bottles clearly differs significantly and would impact your choice should you be dying of thirst.

Fuzzy Conflicts

Tall

Short

Fuzzy Association

Very Tall

Tall

Medium

Short

Very Short
Based on intuition and judgment
No need for a mathematical model
Provides a smooth transition between members and nonmembers
Relatively simple, fast and adaptive
Less sensitive to system fluctuations
Can implement design objectives, difficult to express mathematically, in linguistic or descriptive rules.

Some Interesting Applications
- Smooth ride control
- Camcorder auto-focus and jiggle control
- Braking systems
- Copier quality control
- Rice cooker temperature control
- High performance drives
- Air-conditioning systems

Why Fuzzy Logic
- Based on intuition and judgment
- No need for a mathematical model
- Provides a smooth transition between members and nonmembers
- Relatively simple, fast and adaptive
- Less sensitive to system fluctuations
- Can implement design objectives, difficult to express mathematically, in linguistic or descriptive rules.

Applications Domain
- Fuzzy Logic
- Fuzzy Modeling
  - Neuro-Fuzzy System
- Fuzzy Control
  - Intelligent Control
  - Hybrid Control
- Fuzzy Pattern Recognition

Swarm Intelligence

Swarm Intelligence
Swarm Intelligence

Coordination without Direct Communication

Swarm Intelligence

- Appears in swarms of certain insect species
- Interactions is indirect (stigmergy)
- Complex forms of social behavior can achieve a number of tasks

Features

- Distributed and Multi-agent search
- Uses stigmergy (communication through the environment) for agent interaction
- Adaptive
- Uses reinforcement learning
- Randomness → robustness

SWIN - Stigmergy

- Stigmergy means communication through the environment
  - Pheromone laying: pheromone attracts ants who lay even more pheromone
  - Task-related stigmergy: e.g. termite nest building

Example of Swarm Intelligence

Application to Complex Systems (Routing in Data Networks)
Routing in Data Networks

Why is it so complex?
- Distributed problem, requires coordination of a number of nodes, which might not have direct communication
- Must cope with node failures
- Must re-adjust routes that get congested

Existing Schemes

Wired
- Distance vector (Bellman-Ford)
- Linkstate (Dijkstra’s)
- OSPF (Internet Standard – linkstate-based)

Wireless
- Table-driven (all nodes have routing tables to all destinations)
- On-demand (routing tables are created on demand)

Swarm Advantages in Routing

<table>
<thead>
<tr>
<th>Swarm-based</th>
<th>Non-Swarm-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Some of them(e.g Bellman Ford)</td>
</tr>
<tr>
<td>Adaptive</td>
<td>Yes, can be very fast</td>
</tr>
<tr>
<td></td>
<td>Some, but may cause oscillations. Can be slow.</td>
</tr>
<tr>
<td>Robustness</td>
<td>Yes, packets can be re-routed if links fail</td>
</tr>
<tr>
<td></td>
<td>Yes for distributed BF, but not for link-state.</td>
</tr>
<tr>
<td>Load balancing</td>
<td>Yes, adapts to congestion patterns</td>
</tr>
<tr>
<td></td>
<td>Slow adaptation</td>
</tr>
</tbody>
</table>

AntNet

- Introduced by Di Caro and Dorigo
- Designed for packet – switching networks
- Uses ants who affect routing based on trip times to destination
- This is not done directly, but through a processed quantity

ANTNET: Routing Tables

<table>
<thead>
<tr>
<th>Current</th>
<th>B</th>
<th>F</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>0.28</td>
<td>0.72</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Sum of row=1

Forward Ants

AB 0.22
BC 0.11
CD 0.14
DE 0.15
AC 0.11
BC 0.11
DE 0.23
The Algorithm-Essential steps

- **Launch** forward ant
- Find path **randomly**, but based on routing table probabilities
- Remember the path and the arrival times for forward ant
- Backward ants trace path in reverse
- While at that, they **update** routing tables

Information Carried by Backward Ant

- **Trip-time table**

<table>
<thead>
<tr>
<th>Destination</th>
<th>Mean</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>0.24</td>
<td>0.02</td>
</tr>
<tr>
<td>G</td>
<td>0.18</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Contains trip-time estimate to every destination

AntNet: Table updates

\[
P_{df} = P_{df} + \Delta_\

P_{dh} = P_{dh} + \Delta_-

\Delta_+ = (1 - r') (1 - P_{df})

\Delta_- = - (1 - r') P_{dh}, \ n \neq f, \ n \in N
\]

- \( P_{df} \) is the probability of choosing node \( f \) as next-hop, when the destination is \( d \)
- \( N \) is the set of neighbors of the current node
- \( f \) is the next-hop node
- \( n \) are all the other neighbor nodes
- \( \Delta_\) and \( \Delta_- \) are the positive and negative reinforcement, respectively
- \( r' \) = weighting factor (step size)

Why Forward and Backward

- **Routing tables** are updated only by ants who were **successful** in reaching destination
- **If a ant is lost**, it will not generate a backward ant next-hops that were not successful will not be updated

Weighting Factor

\[
r' = \begin{cases} \frac{T}{c\mu}, & c \geq 1, \text{ if } \frac{T}{c\mu} < 1 \\ 1, & \text{otherwise} \end{cases}
\]

- \( r' \) = weighting factor
- \( T \) = trip time from current node to the destination
- \( \mu \) = average of \( T \)
- \( c \) = scaling factor (usually 2)
AntNet

- Equation are based on negative feedback:
  - Positive reinforcement ($\Delta^+$) should be smaller as probabilities get larger and vice versa
  - Negative reinforcement ($\Delta^-$) should be larger as probabilities get larger and vice versa

Problem: $T$ could be unreliable (high variance)
Solution: Use $\sigma$ over $\mu$ ratio to adjust $r'$
Why? Because high $\sigma/\mu$ means $T$ is unreliable and vice versa

Adjusting $r'$

<table>
<thead>
<tr>
<th>Good time</th>
<th>Bad time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r' = T/\mu &lt; 0.5$</td>
<td>$r' = T/\mu &gt; 0.5$</td>
</tr>
<tr>
<td>$T$ unreliable</td>
<td>$T$ reliable</td>
</tr>
<tr>
<td>$\sigma &gt; \delta$</td>
<td>$\sigma &lt; \delta$</td>
</tr>
<tr>
<td>$-\frac{a\sigma}{\mu}$</td>
<td>$-\frac{a\sigma}{\mu}$</td>
</tr>
<tr>
<td>$1 - e^{-\frac{\sigma}{\mu}}$</td>
<td>$1 - e^{-\frac{\sigma}{\mu}}$</td>
</tr>
</tbody>
</table>

- Add nonlinearity: $r' \rightarrow (r')^h$
- Why? To manipulate the learning rate $r'$
- Choice of $h$: usually 2 (heuristic)

Objective

- To move a group of vehicles (rovers, robots, underwater vehicles) independently to achieve common or distributed objectives.

Wireless Issues

- Broadcast advantage
  - When node transmits, it can be heard by all nodes in its range
- Energy constraints
  - Routing must take into account energy
  - Energy can also be adjusted by adjusting range
- Noise and bandwidth requirements also affect energy and range
- Node mobility: nodes can move in and out of range
**Artificial Potential Field (APF)**

- The motion of the vehicle is governed by a minimum of two fields: attractive and repulsive fields.
- **Attractive field:** a force that moves the vehicle toward the target position.
- **Repulsive field:** a force that moves the vehicle away from obstacles or areas that is being covered by other vehicles.

**Attraction Velocity**

Cartesian distance between the vehicle and the target

\[
V_a = D_a \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}
\]

\(D_a\): distance between the vehicle and the target
\(\theta\): angle between the vehicle and the target

**Repulsion Velocity**

\[
V_r = \frac{1}{D_r} \begin{bmatrix} \cos \gamma \\ \sin \gamma \end{bmatrix}
\]

\(D_r\): distance between the vehicle and the obstacle.
\(\gamma\): angle between the vehicle and the obstacle.

**Total Velocity**

\[
\vec{V} = \alpha \vec{V}_a + \beta \vec{V}_r + \zeta
\]

- \(\alpha\) and \(\beta\) are modulation parameters (can be dynamically changing)
  - For example: if the vehicle is to achieve a stand-still state at the target point, \(\beta\) is set to zero when \(D_a\) approaches zero.
- \(\zeta\) is the effect of other influences (other robots to avoid)

**Boids**

- Image of a flock of birds flying and avoiding obstacles.
- Text: "Boids"
History of Boids (Flocking)

- Pioneered by Craig Reynolds
- Coherent flocking behavior using simple rules was first demonstrated in the late 1980s.
  - To develop realistic motions of groups of ‘actors’ in computer animation.
  - Each actor (boid) has a scripted path predetermined by the animator.

Flocking rules

- Basic rules (necessary and sufficient)
  - Cohesion: fly towards the center of your local flock mates.
  - Separation: keep a certain distance away from nearest flock mates.
  - Alignment: align your velocity vector with that of the local flock.
- Additional rules (for better flocking dynamics)
  - Evasion: avoid occupying the same local airspace as your nearest flockmate. Evasion is a localized form of separation.
  - Migration: fly towards a pre-specified location

Cohesion

- Move towards the center of local flockmates

Separation

- Steer to avoid collision with local flockmates

Alignment

- Steer towards the average heading of local flockmates

Evasion

- Avoid occupying the same local space as your nearest flockmate.
Migration

- Fly towards a pre-specified location

Final motion vector

Original velocity

Net velocity

Alignment

Cohesion

Evasion

Mitigation

Separation

Control Applications

- Simple behaviors for individuals and pairs:
  - Seek and Flee
  - Pursue and Evade
  - Obstacle Avoidance
  - Path or wall Following
  - Flow Field Following

- Combined behaviors and groups:
  - Crowd Path Following
  - Leader Following
  - Unaligned Collision Avoidance
  - Queuing (at a doorway)
  - Flocking (combining: separation)

Particle Swarm Optimization

Coordination with Direct Communication
Particle Swarm Optimization

- **Inventors:** James Kennedy and Russell Eberhart
- An Algorithm originally developed to imitate the motion of a Flock of Birds, or insects
- Assumes Information Exchange (Social Interactions) among the search agents
- **Basic Idea:** Keep track of
  - Global Best
  - Self Best

How does it work?

- **Problem:**
  Find X which minimizes f(X)
- **Particle Swarm:**
  - **Start:** Random set of solution vectors
  - **Experiment:** Include randomness in the choice of new states.
  - **Remember:** Encode the information about good solutions.
  - **Improvise:** Use the ‘experience’ information to initiate search in a new regions

---

Current motion

Component in the direction of global best

Component in the direction of personal best

Component in the direction of previous motion

New Motion

Global best

Personal Best at previous step
**PSO Modeling**

- Each solution vector is modeled as
  - The coordinates of a bird or a ‘particle’ in a ‘swarm’ flying through the search space
  - All the particles have a non-zero velocity and thus never stop flying and are always sampling new regions.
- Each ‘particle’ remembers
  - Where the global best and where the local best are.

**The search is guided by**

- The collective consciousness of the swarm
- Introducing randomness into the dynamics in a controlled manner

**Particle Swarm Dynamics**

\[
\begin{align*}
\dot{x}(k+1) &= \dot{x}(k) + \dot{v}(k) \\
\dot{v}(k+1) &= w\dot{v}(k) + r(0,a_1)(x_{	ext{selfBest}}(k) - \bar{x}(k)) \\
&\quad + r(0,a_2)(x_{	ext{groupBest}}(k) - \bar{x}(k))
\end{align*}
\]

- Inertia
- Control randomness

**PSO**

- \( x \) is a solution vector ‘particle’ and \( v \) is the velocity of this particle
- \( a_1 \) and \( a_2 \) are two scalars,
- \( w \) is the inertia
- \( r(0,1) \) is a uniform random number generator between 0 and 1

**Design Parameters**

- \( a_1 \) and \( a_2 \)
- \( w \): Should be between [0.9 and 1.2]
  - High values of \( w \) gives a global search
  - Low values of \( w \) gives a local search
- \( v_{\text{max}} \): To be designed according to the nature of the search surface.

**The Art of Fitness Function**

- To move robots to boundary points
  - Metric: \(|f(x) - \text{boundary value}|\)
The Art of Fitness Function

- Distribute robots uniformly on the boundary close to current state

Metric:
\[ |f(x) - \text{boundary value}| - \text{Distance to closest neighbor} + \text{Distance to current state} \]

(penalize proximity to neighbors, penalize distance from current state)

PSO Challenges

- Like any search technique, PSO could be unsuccessful at distinguishing between global and local minima
  - Local minimum is easier to find
  - If fitness function cannot amplify the difference between global and local minima, PSO is likely to stay in the local minima
**Modified PSO**

- Two-step PSO (or gradient-approximation)
- Cluster PSO

**Two-Step PSO**

- Each particle takes two steps: short and long
  - Then decide on optimal step based on steeper negative gradient

**Cluster PSO**

- It is a hierarchical version of PSO:
  - PSO are arranged in clusters
  - Each cluster contains multiple agents
  - Each cluster has a centroid that acts, effectively, as a standard PSO agent
  - Each agent within the cluster is attracted to its personal best, the cluster best, and the cluster centroid

**Two-Step PSO**

- Better method at not “overflying” narrow valleys
- **Problems:**
  - Particles may take the short step more often than the long step, resulting in slower convergence
  - Can still get trapped in local minima

**Cluster PSO**

\[
\begin{align*}
v_c &= w_c v_c + a_{c1} (x_{cb} - x_c) + a_{c2} (x_{gb} - x_c) \\
v_a &= w_a v_a + a_1 (x_{ab} - x_a) + a_2 (x_{cb} - x_a) + a_3 (x_{cc} - x_a) \\
x_{cc} &= x_{cc} + v_c \\
x_a &= x_a + v_a + v_c
\end{align*}
\]
**Cluster PSO**

- Cluster PSO combines globally superior ability of standard PSO in avoiding local minima with the locally efficient search which can find narrow global minima.
- MAYBE!

**Conventional Optimization**

- Most optimization problems have several (possibly conflicting) objectives.
- These problems are frequently treated as
  - Single-objective optimization problems by transforming all but one objective into constraints.
  - Single cost function where all objectives are weighted according to their importance.

**Multiobjectives Optimization & Pareto Fronts**

**Weighted Aggregation for Multi-Objective Optimization**

- Objective: minimize the following functions
  \[ f_1(x), f_2(x), \ldots, f_n(x) \]
- Formulation: aggregate all functions in a weighted fashion to form one cost index
  \[ J(x) = w_1 f_1(x) + w_2 f_2(x) + \ldots + w_n f_n(x) \]

**Conventional Optimization**

- Any optimization problem can be framed as finding the parameter minimizing a given objective function
- find \( x' \) that minimizes \( y = f(x) \)
- Typically, one solution

**Limitation of Aggregated Optimization function**

- Single aggregated function leads to only one solution.
- The relative importance must be selected before hand and the final solution is dependent on the selection.
- Trade-offs can not be easily evaluated.
- Solution is not possible for nonconvex search spaces.
**Trade-off Analysis (Example of Conflicting Objectives)**
- Designing of **distributed controllers** while reducing the **cost**.
- Place functional blocks on a chip such that **chip area is minimized** and **power dissipation** is also minimized.
- Find the vehicle that covers the **most distance** in a day while requiring the **least energy**.
- Placing **enough sensors** to monitor a system while reducing the **cost**.

**Multi-Objective Optimization: Pareto Front**

**Goal of Multi-Objective Optimization (MOO)**
- Finding a set of decision variables that
  - satisfies constraints and
  - optimizes a vector of objectives.
- The term “optimize” means finding a solution which would give values of all the objective functions acceptable to the designer.

**Simple Problem, Single solution**

Find \( x^* \) such that \( f_i(x^*) \leq f_i(x) \) for all \( i \)

**Simple Problem, Multiple Solutions**

**Example: Operational Amplifier**

**Mathematical Formulation of MOO**
- Find the vector \( x^* = [x_1^*, x_2^*, ..., x_n^*]^T \)
  which satisfies
  - \( m \) inequality constraints:
    \[ g_i(x) \geq 0, \quad i = 1, 2, ..., m \]
  - and \( p \) equality constraints:
    \[ h_i(x) = 0, \quad i = 1, 2, ..., p \]
  - and optimizes the vector function:
    \[ f(x) = [f_1(x), f_2(x), ..., f_k(x)]^T \]
**Terminologies used in MOO**

- The solution is dominant if
  \[ f_i(x^*) \leq f_i(x) \text{ for all } i, \text{ and} \]
  \[ f_i(x^*) < f_i(x) \text{ for at least one } i \in \{1, 2, \ldots, k\} \]

- The solution is non-dominant if no more dominant solution is found
  - The search is converged to one solution (Pareto optimal point)

**Pareto Front Algorithm**

- Compute Constraints
- Compute Objectives
- PSO, EC, ...
- Delay
- Test
- Test
- Pareto Optimal Point
- Infeasible
- Dominant
- Feasible Solution

**Example: Transportation Problem**

- **Distance**
- **Energy**

**The Future**

- Time
- Interest
- Hype
- Backlash
- Serious Development
- Implementations
"(NN, Fuzzy, ...) has no place in scientific literature."

"Already, primitive Intelligent Agents are buzzing around on the internet."
Time Magazine, March 25, 1996, P.57

"This telephone has too many shortcomings to be seriously considered as a means of communication. The device is inherently of no value to us."
Western Union internal memo, 1876.

"The wireless music box (radio) has no imaginable commercial value. Who would pay for a message sent to nobody in particular?"
David Sarnoff’s Associates in response to his urgings for investment in the radio in 1920’s
Paradigm Shifts (Business)

“We don’t like their sound, and guitar music is on the way out.”
Decca Recording Company rejecting the Beatles in 1962.

Paradigm Shifts (Computing)

“640 K ought to be enough for anybody.”

Paradigm Shifts (Science)

“Louis Pasteur’s theory of germs is ridiculous fiction”
Pierre Pachet, Professor at Toulouse, 1872.

“Airplanes are interesting toys but of no military value”
Marechal Ferdinand Foch, Professor of Strategy, Ecole Superieure de Guerre.

Paradigm Shifts (Computers)

“I think there is a world market for maybe five computers.”
Thomas Watson, Chairman of IBM, 1943.

“But what is it good for?”

Remarks

- BIA is the wave of the future.
  - Simplicity and power of BIA
- The key test is in the applications.
- Several industrial applications and consumer products are equipped with BIA based systems

Real Test

- When safety is a concern, BIA may not be a stand-alone system.
  - Stability, reliability and availability of BIA can not, so far, be verified by a closed form mathematics.
  - If the control breakdown results in safety hazard, or may lead to expensive repair
- For these systems, classical systems is probably the method of choice by most engineers.
The Cranial Nerves

Olfactory Nerve
Optic Nerve
Oculomotor Nerve
Trigeminal Nerve
Abducens Nerve
Trochlear Nerve
Facial Nerve
Auditory Nerve
Glossopharyngeal
Vagus Nerve
Hypoglossal Nerve
Accessory Nerve

Finis